CSE 503
Introduction to Computer Science for Non-Majors

Dr. Eric Mikida
epmikida@buffalo.edu
208 Capen Hall

Day 29
Reading and Debugging Code

Announcements

- Lab #4 due tonight at 11:59pm
- Lab #5 will be released ASAP

Recap

- Last time we started looking at the MusicRater web app
  - It is a larger application than we are used to, with 4 different files, many functions, and some new things we have not seen yet
  - 2 version (so far): 1.0 uses CSV files to store data, and 2.0 uses DB
- Being able to get at least a baseline understanding of what is happening by reading code is an important skill to have
  - Even if you don't know exactly what it is doing, you should be able to at least make a reasonable guess, and formulate targeted questions

Read through MusicRater a bit more…
Ask questions about the code!!!

MusicRater Code

Four main files:

1. main.py: The Python server code, uses bottle
2. index.html: The code for the webpage…looks surprisingly empty
3. myCode.js: Makes the webpage interactive **AND** generates HTML
4. ratings.py: Reads and Writes the persistent data to files (CSV or DB)

New Things

Other than the amount of code, the new things are very few in number:

1. Hyperlink HTML elements: `<a href="www.buffalo.edu">UB</a>`
2. Adding new HTML elements to an existing file (see generate_song_html and generate_button_html in myCode.js)

Now what if something goes wrong?

Previously, we added **`print`/`console.log`** statements to our code to track what it was doing in case things went wrong

Those techniques still apply…but with bigger projects it can be hard to know where to start

Now what if something goes wrong?

Previously, we added **`print`/`console.log`** statements to our code to track what it was doing in case things went wrong

Those techniques still apply…but with bigger projects it can be hard to know where to start

Like with many things we've discussed…don't dive right into the code!

Now what if something goes wrong?

At this point you are like a detective...you have some clues (things aren't behaving like you expect), and you need to find the cause

1. **Gather up your evidence first:** What is the symptom that you noticed, what pieces of information do you already have (console output)
- **Start thinking big picture first:** Think conceptually about what different pieces of functionality exist, and how they interact
- **Think big to small:** Think about files first, then after you've narrowed things down think about functions, then about individual lines